DAY 3- API Integration Report-[Nike E-Commerce]

-Project creation at Sanity

```
Your content will be stored in a dataset that can be public or private, depending on whether you want to query your content with or without authentication.

The default dataset configuration has a public dataset named "production".

Use the default dataset configuration? Yes

Creating dataset

Would you like to add configuration files for a Sanity project in this N ext.js folder? Yes

It looks like you are using Next.js 15 and React 19

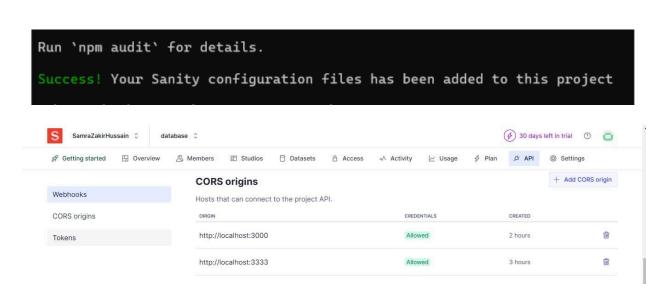
Please read our compatibility guide.

https://www.sanity.io/help/react-19

Would you like an embedded Sanity Studio? Yes

Would you like an embedded Sanity Studio? /studio

Select project template to use Clean project with no predefined schema t
```



-Adjustments made to schemas and Data migration

```
| Sanity | SchemaTypes | Sanity | SchemaTypes | T8 products | Sanity | SchemaTypes | T8 products | Sanity | SchemaTypes | T8 products | Sanity | SchemaTypes | T8 product | T8 product
```

```
OPEN EDITORS
                                                     import { createClient } from '@sanity/client';
import fetch from 'node-fetch';
DAY-3-HACKATHON
                              日日日日
 layout.tsx
                                                           const client = createClient({
  projectId: "53117kxa",
 page.tsx
                                                            useCdn: false, // Set to true if you want faster reads
apiVersion: '2025-01-13',
token: "skEpUcwFmLmntmgtE0auguUWnKa0c9KSs0Cf7gCOZSicNQI2536id975ACU76U1AiL6eszOf6U5xTqeZ99lrBdiQyFWT6GwgYJ
                                          U
                                                           // Function to upload an image to Sanity
async function uploadImageToSanity(imageUrl) {
                                                               console.log(`Uploading image: ${imageUrl}`);
eslintrc.ison
.gitignore
                                                             const response = await fetch(imageUrl);
if (!response.ok) {
                                                              throw new Error('Failed to fetch image: ${imageUrl}');
}
TS next.config.ts
{} package-lock.json
JS postcss.config.mjs
                                                               const buffer = await response.arrayBuffer();

    README.md
```

```
// Function to fetch products from the provided API and upload them to Sanity
async function migrateProducts() {
   try {
      const response = await fetch('https://template-0-beta.vercel.app/api/product');
      if (!response.ok) {
            throw new Error('HTTP error! Status: ${response.status}');
      }
      const products = await response.json();
      for (const product of products) {
            await uploadProduct(product);
      }
      } catch (error) {
      console.error('Error fetching products:', error);
    }
}

// Start the migration
migrateProducts();
```

-Populated Sanity CMS fields

